Probabilistic Graphical Models

Brown University CSCI 2950-P, Spring 2013
Prof. Erik Sudderth

Lecture 8:
Inference & Learning for Exponential Families,
Expectation Maximization (EM) Algorithm

Some figures courtesy Michael Jordan’s draft textbook,
An Introduction to Probabilistic Graphical Models
Exponential Families of Distributions

\[ p(x|\theta) = \frac{1}{Z(\theta)} h(x) \exp[\theta^T \phi(x)] \]
\[ Z(\theta) = \int_{\mathcal{X}} h(x) \exp[\theta^T \phi(x)] dx \]
\[ = h(x) \exp[\theta^T \phi(x) - A(\theta)] \]
\[ A(\theta) = \log Z(\theta) \]

\( \phi(x) \in \mathbb{R}^d \quad \text{fixed vector of sufficient statistics} \) (features), specifying the family of distributions

\( \theta \in \Theta \quad \text{unknown vector of natural parameters} \), determine particular distribution in this family

\( Z(\theta) > 0 \quad \text{normalization constant or partition function} \), ensuring this is a valid probability distribution

\( h(x) > 0 \quad \text{reference measure} \) independent of parameters (for many models, we simply have \( h(x) = 1 \))

To ensure this construction is valid, we take
\[ \Theta = \{ \theta \in \mathbb{R}^d \mid Z(\theta) < \infty \} \]
Factor Graphs & Exponential Families

\[ p(x) = \frac{1}{Z(\theta)} \prod_{f \in \mathcal{F}} \psi_f(x_f | \theta_f) \]

- \( \mathcal{F} \rightarrow \) set of hyperedges linking subsets of nodes \( f \subseteq \mathcal{V} \)
- \( \mathcal{V} \rightarrow \) set of \( N \) nodes or vertices, \( \{1, 2, \ldots, N\} \)
- \( Z \rightarrow \) normalization constant (partition function)

- A **factor graph** is created from non-negative potential functions
- To guarantee non-negativity, we typically define potentials as

\[
\psi_f(x_f | \theta_f) = \nu_f(x_f) \exp \left\{ \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) \right\}
\]

**Local exponential family:**
\[ \theta_f \triangleq \{ \theta_{fa} | a \in \mathcal{A}_f \} \]

\[
p(x | \theta) = \left( \prod_{f \in \mathcal{F}} \nu_f(x_f) \right) \exp \left\{ \sum_{f \in \mathcal{F}} \sum_{a \in \mathcal{A}_f} \theta_{fa} \phi_{fa}(x_f) - \Phi(\theta) \right\}
\]

\[ \Phi(\theta) = \log Z(\theta) \]
Undirected Graphs & Exp. Families

$$p(x \mid \theta) = \left( \prod_{f \in \mathcal{F}} \nu_f(x_f) \right) \exp \left\{ \sum_{f \in \mathcal{F}} \sum_{a \in A_f} \theta_{fa} \phi_{fa}(x_f) - \Phi(\theta) \right\} \quad \Phi(\theta) = \log Z(\theta)$$

- Pick features to define an exponential family of distributions
- Use factor graph to represent structure of chosen statistics
- Create undirected graph with a clique for every factor node
- **Result:** Visualization of Markov properties of your family
Generalized Linear Models

- General framework for modeling non-Gaussian data with linear prediction, using exponential families:
  - Construct instance-specific natural parameters:
    \[ \theta_i = w^T \phi(x_i) \]
  - Observation comes from exponential family:
    \[ p(y_i \mid x_i, w) = \exp \left\{ y_i \theta_i - A(\theta_i) \right\} \]
- Special cases: linear regression and logistic regression
- ML and MAP estimation is generally straightforward
- Many possible extensions:
  - *Multivariate responses* with more parameters
    (biggest difficulty is notation and indexing)
  - *Link functions* to allow more flexibility in how \( (w, x_i) \rightarrow \theta_i \)
Directed Graphs & Exp. Families

\[ p(x) = \prod_{i=1}^{N} p(x_i \mid x_{\Gamma(i)}, \theta_i) \]

\[ p(x_i \mid x_{\Gamma(i)}, \theta_i) = \exp \left\{ x_i \theta_i^T \phi(x_{\Gamma(i)}) - A(\theta_i^T \phi(x_{\Gamma(i)})) \right\} \]

- For each node, pick an appropriate exponential family
- Pick features of parent nodes relevant to child variable
  *Most generally, indicators for all joint configurations of parents.*
- Child parameters are a (learned) linear func. of parent features
- **Result:** Node-specific generalized linear models
Inference versus Learning

- **Inference:** Given a model with known parameters, estimate or find marginals of “hidden” variables for some data instance
- **Learning:** Given multiple data instances, find (often ML/MAP) estimates of parameters for a graphical model of their structure
  - Training instances may be completely or partially observed

*Example: Expert systems for medical diagnosis*

- **Inference:** Given observed symptoms for a particular patient, infer probabilities that they have contracted various diseases
- **Learning:** Given a database of many patient diagnoses, learn the relationships between diseases and symptoms

*Example: Markov random fields for semantic image segmentation*

- **Inference:** What object category is depicted at each pixel?
- **Learning:** How do objects relate to low-level image features?
Mean Parameter Spaces

\[ p(x \mid \theta) = \exp\{\theta^T \phi(x) - A(\theta)\} \]

\[ \mu_a = \mathbb{E}_p[\phi_a(x)] = \int \phi_a(x)p(x) \, dx \]

\[ \mathcal{M} \triangleq \{ \mu \in \mathbb{R}^d \mid \exists p \text{ such that } \mathbb{E}_p[\phi(x)] = \mu \} \]

- For a given collection of sufficient statistics, what is the set of all \textit{realizable} mean parameters?

**Scalar Gaussian**

**Pair of Binary Variables**

- The set of realizable parameters is always \textit{convex}. Why?
Inference and Learning

3.6 Conjugate Duality: Maximum Likelihood and Maximum Entropy

3.6.2 Some Simple Examples

Theorem 3.4 is best understood by working through some simple examples. Table 3.2 provides the conjugate dual pair \((A, A^*)\) for several well-known exponential families of scalar random variables. For each family, the table also lists \(\Omega := \text{dom } A\), as well as the set \(M\), which contains the effective domain of \(A^*\), corresponding to the set of values for which \(A^*\) is finite.

In the rest of this section, we illustrate the basic ideas by working through two simple scalar examples in detail. To be clear, neither of these examples is interesting from a computational perspective — indeed, for most scalar exponential families, it is trivial to compute the mapping between canonical and mean parameters by direct methods. Nonetheless, they are useful in building intuition for the consequences of Theorem 3.4. The reader interested only in the main thread may skip ahead to Section 3.7, where we resume our discussion of the role of Theorem 3.4 in the derivation of approximate inference algorithms for multivariate exponential families.

Example 3.10 (Conjugate Duality for Bernoulli).

Consider a Bernoulli variable \(X \in \{0, 1\}\): its distribution can be written as an exponential family with \(\phi(x) = x\), \(A(\theta) = \log(1 + \exp(\theta))\), and \(\Omega = \mathbb{R}^d\). In order to verify the claim in Theorem 3.4(a), let us compute the conjugate dual function \(A^*\) by direct methods. By the definition of conjugate
\[
\mu_a = \mathbb{E}_p[\phi_a(x)] = \int \phi_a(x)p(x) \, dx
\]
\[
\mathcal{M} \triangleq \{\mu \in \mathbb{R}^d \mid \exists \, p \text{ such that } \mathbb{E}_p[\phi(x)] = \mu\}
\]
Supervised Learning

Generative ML or MAP Learning:

\[
\max_{\pi, \theta} \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} [\log p(y_i | \pi) + \log p(x_i | y_i, \theta)]
\]

Discriminative ML or MAP Learning:

\[
\max_{\theta} \log p(\theta) + \sum_{i=1}^{N} \log p(y_i | x_i, \theta)
\]
Unsupervised Learning

Clustering:

\[
\max_{\pi, \theta} \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[ \sum_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \right]
\]

Dimensionality Reduction:

\[
\max_{\pi, \theta} \log p(\pi) + \log p(\theta) + \sum_{i=1}^{N} \log \left[ \int_{z_i} p(z_i \mid \pi) p(x_i \mid z_i, \theta) \, dz_i \right]
\]

- No notion of training and test data: labels are never observed
- As before, maximize posterior probability of model parameters
- For hidden variables associated with each observation, we marginalize over possible values rather than estimating
  - Fully accounts for uncertainty in these variables
  - There is one hidden variable per observation, so cannot perfectly estimate even with infinite data
- Must use generative model (discriminative degenerates)
Unsupervised Learning Algorithms

- **Initialization**: Randomly select starting parameters
- **Estimation**: Given parameters, infer likely hidden data
  - Similar to *testing* phase of supervised learning
- **Learning**: Given hidden & observed data, find likely parameters
  - Similar to *training* phase of supervised learning
- **Iteration**: Alternate estimation & learning until convergence

\[
\pi, \theta \quad \rightarrow \quad \text{parameters (shared across instances)}
\]

\[
\pi, \theta \quad \rightarrow \quad \text{hidden data (unique to particular instances)}
\]
Expectation Maximization (EM)

**Supervised Training**
- $\pi, \theta$
- $z_1, \ldots, z_N$

**Supervised Testing**
- $\pi, \theta$

**Unsupervised Learning**
- $\pi$

- **Initialization**: Randomly select starting parameters
- **E-Step**: Given parameters, find posterior of hidden data
  - Equivalent to test inference of full posterior distribution
- **M-Step**: Given posterior distributions, find likely parameters
  - Distinct from supervised ML/MAP, but often still tractable
- **Iteration**: Alternate E-step & M-step until convergence

Parameters (shared across observations)

Hidden data (unique to particular instances)
Concavity & Jensen’s Inequality

\[ \ln(\mathbb{E}[X]) \geq \mathbb{E}[\ln(X)] \]
### EM as Lower Bound Maximization

\[
\ln p(x \mid \theta) = \ln \left( \sum_z p(x, z \mid \theta) \right)
\]

\[
\ln p(x \mid \theta) \geq \sum_z q(z) \ln \left( \frac{p(x, z \mid \theta)}{q(z)} \right)
\]

\[
\ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta)
\]

- **Initialization:** Randomly select starting parameters \( \theta^{(0)} \)
- **E-Step:** Given parameters, find posterior of hidden data

\[
q^{(t)} = \arg \max_q \mathcal{L}(q, \theta^{(t-1)})
\]

- **M-Step:** Given posterior distributions, find likely parameters

\[
\theta^{(t)} = \arg \max_\theta \mathcal{L}(q^{(t)}, \theta)
\]

- **Iteration:** Alternate E-step & M-step until convergence
Lower Bounds on Marginal Likelihood

\[ \text{E-Step:} \quad q(z) = p(z \mid x, \theta) \]

\[ \text{KL}(q \mid \mid p) \]

\[ \mathcal{L}(q, \theta) \]

\[ \ln p(X \mid \theta) \]

C. Bishop, Pattern Recognition & Machine Learning
EM: Expectation Step

\[ \ln p(x \mid \theta) \geq \sum_z q(z) \ln p(x, z \mid \theta) - \sum_z q(z) \ln q(z) \triangleq \mathcal{L}(q, \theta) \]

\[ q^{(t)} = \arg \max_q \mathcal{L}(q, \theta^{(t-1)}) \]

- General solution, for any probabilistic model:
  \[ q^{(t)}(z) = p(z \mid x, \theta^{(t-1)}) \]

- For a directed graphical model:
  - \( \theta \) fixes conditional distributions of every child node, given parents
  - \( x \) observed nodes (training data)
  - \( z \) unobserved nodes (hidden data)

Inference: Find summary statistics of posterior needed for following M-step